

**IBM MAINFRAME**

**INSTRUKTIONS-  
PROCESSOREN**

**&**

**MASKINENS INTERNE LAGRE**

# INSTRUKTIONSPROCESSOREN.

**Eksempel:**

**Programsekvens:**

**Sourceprogram:**

```

read INDATA into INFELT.
if NR = 40 then do:
    OUTFELT = INFELT.
    LISTE = NR.
else do:
    OUTFELT = INFELT.
    LISTE = 10.
write OUTDATA from OUTFELT

```

**JCL-sekvens:**

```
//STP EXEC PGM=progsekv
//INDATA DD DSN=OLDREG,UNIT=DISK,...
//OUTDATA DD SYSOUT=A
```

Load-modul:

```

read 200 BYTES into STORAGE-ADR 2B+200
if STORAGE-ADR 2D+2 = 40 then do:
    STORAGE-ADR 7G+200 = STORAGE-ADR 2B+200
    STORAGE-ADR 6A+2 = STORAGE-ADR 2B+14
else do:
    STORAGE-ADR 7G+200 = STORAGE-ADR 2B+200
    STORAGE-ADR 6A+2 = 10
write 132 bytes from STORAGE-ADR 7G+132

```

Storageindhold:

[illegible]

## INSTRUKTIONSPROCESSOREN.

Instruktionen skal principielt gennem 4 faser:

### 1. INSTRUCTION FETCH

hvor selve instruktionen hentes fra sin plads i det interne lager op i et særligt register.

### 2. INSTRUCTION DECODING.

OP-CODE'en identificeres og på grundlag af instruktionens art beregnes adressen på den næste instruktion i programmet. Adressen placeres i CPSW'et sidste bytes.

### 3. OPERAND FETCHING.

Operandernes adresser oversættes mellem virtuelle og reelle adresser, hvorefter data hentes og placeres i registre el. lign. i forhold til instruktionens type og formål.

### 4. INSTRUCTION EXECUTION.

Instruktionen udføres; data bearbejdes og skrives tilbage i sine oprindelige lageradresser.

For at kunne klargøre den næste instruktion må maskinen vide hvilken sker ved at vise adressen på hvor i lageret den ligger placeret.

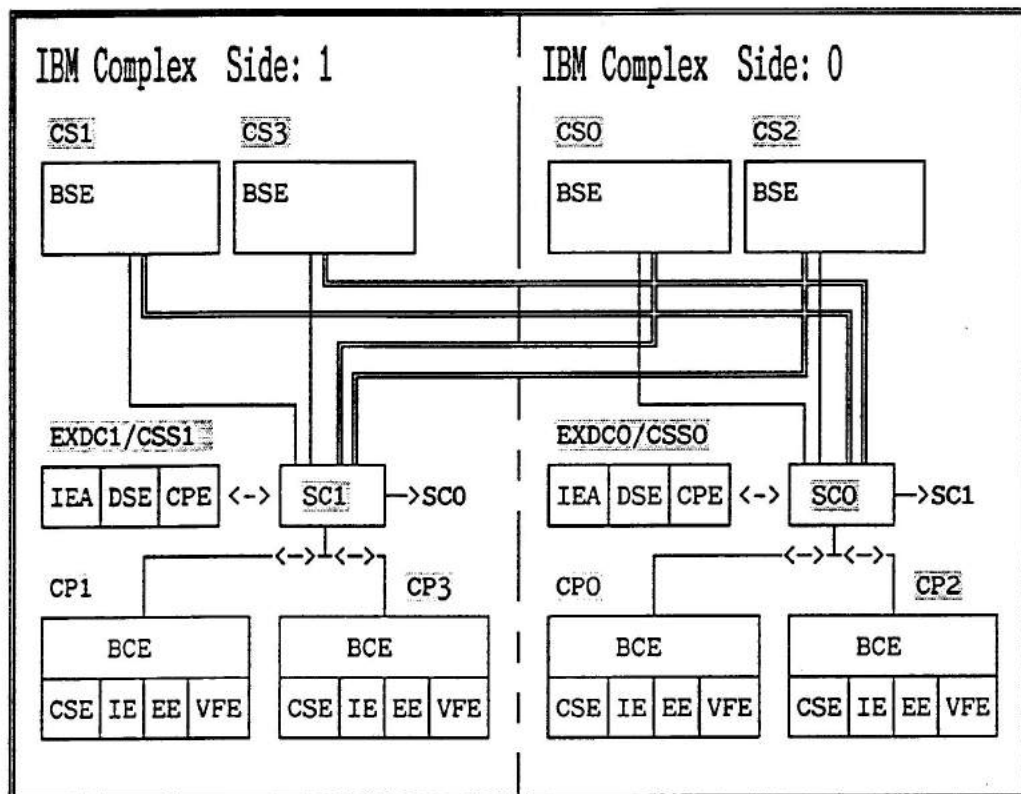
Denne styring sker i CPSW'et i 2. fase (Current Program Status). Adressen beregnes udfra den aktuelle instruktions egen længde (fylde), hvorved den næste logisk ligger placeret umiddelbart efter.

Fysisk kan det se ganske anderledes ud, men det styres af tabeller og kontrolblokke i systemet.

De mellemstore til helt store maskiner arbejder med overlappende instruktionseksekvering, hvor de 4 faser arbejder parallelt med hver sin instruktion, men kun en ad gangen bliver udført.

## INSTRUKTIONSPROCESSOREN.

Nedenstående tegning skal udelukkende betragtes som en PRINCIP-SKITSE for hvordan processor-complexer er opbygget og hvordan de forskellige elementer arbejder sammen.



Forkortelserne har følgende betydninger:

**CSx** Central Storage Element  
BSE Basic Storage Element

Det interne lager

**EXDCx** EXternal Data Controller  
IAEx Interface Adaptor Element  
DSEx Data Server Element  
CPEx Channel Processor Element

Datkanaler

**SCx** System Controller

System kontrol

**CPx** Central Processor

Instruktionsprocessor

BCE Buffer Control Element  
CSE Control Storage Element  
IE Instruction Element  
EE Execution Element  
VFE Variable Field Element

## INSTRUKTIONSPROCESSOREN.

Hver af 3081, 3084 og 3090's Central Processor's er micro-kode kontrolleret og indeholder et Instruction-Element (IE), Execution-Element (EE), Control-Storage-Element (CSE) og et Buffer-Control-Element (BCE).

CSE henter micro-kode instruktioner som kontrollerer udførelsen af instruktioner i IE og EE. BCE kontrollerer overførslen af data mellem central storage og den Central Processor som indeholder BCE'end. Dynamisk adresseoversættelse er en automatisk funktion i BCE'en.

Mere detaljeret indeholder og udfører:

### BCE - Buffer Control Element:

- Håndterer alle Central Processorens referencer til og fra Central Storage.
- Indeholder High-Speed-Bufferen
- Indeholder et buffer Directory
- Indeholder Translation Lookaside Buffer (TLB)
- Udfører Dynamic Address Translation via DAT-hardware.

### I-Element (Instruction Element):

- Henter instruktioner
- Decoder instruktioner
- Beregner operandadresser
- Kø'er instruktioner til E-Elementet
- Henter operander til E-Elementet.
- Processer flere instruktionsstrømme på samme tid.

### E-Element (Execution Element):

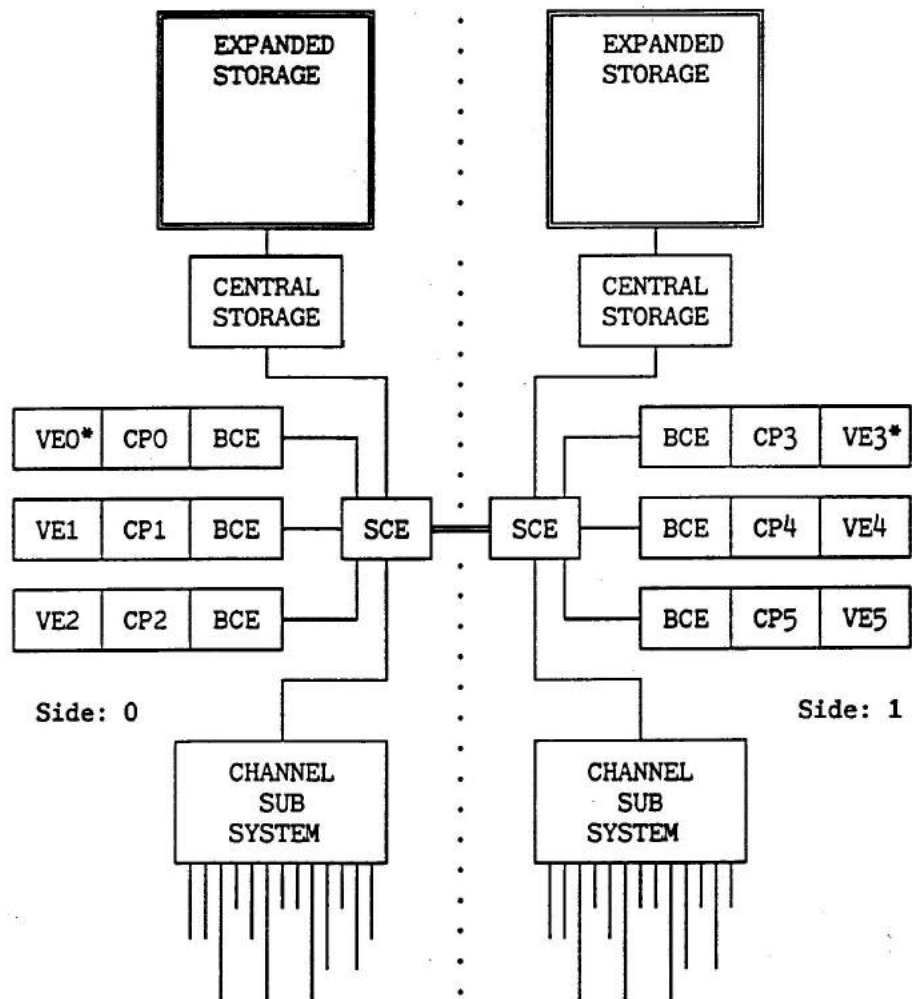
- Henter instruktioner fra kø
- Eksekverer alle instruktioner
- Udfører logiske beslutninger
- Udfører aritmetiske funktioner
- Lagrer resultaterne.
- E-elementets arbejde overlapper I-elementets for og øge effektiviteten.

Bem:

Der er mindre forskelle i den anvendte terminologi på en IBM 308x-, en IBM 3090- og en ES/9000-maskine, bla. hvad angår ovennævnte emner, men principperne er stort set de samme.

# MASKINENS INTERNE LAGRE.

## EXPANDED STORAGE / Traditionel placering:



# MASKINENS INTERNE LAGRE.

EXPANDED STORAGE / ES/9000 Største modeller.

